

With-defaults Capability for NETCONF

Свойство :with-defaults для протокола NETCONF

Аннотация

Протокол настройки сети (Network Configuration Protocol или NETCONF) определяет способы считывания и редактирования данных конфигурации с сервера NETCONF. В некоторых случаях часть таких данных клиент NETCONF может не устанавливать, используя взамен принятые по умолчанию значения, известные серверу. Во многих случаях клиент NETCONF заранее знает принятые по умолчанию значения и серверу NETCONF не требуется записывать их в хранилище конфигурации NETCONF или передавать клиенту в отклике на операцию извлечения данных (retrieval). В иных ситуациях клиенту NETCONF будут требоваться данные от сервера. Не все реализации серверов одинаково трактуют принятые по умолчанию данные. Этот документ определяет основанное на возможностях расширение протокола NETCONF, позволяющее клиентам NETCONF определить, как сервер обрабатывает принятые по умолчанию значения, а также задаёт новые механизмы для управления клиентом обработкой принятых по умолчанию данных на сервере.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF¹ и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG². Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 5741.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <http://www.rfc-editor.org/info/rfc6243>.

Авторские права

Авторские права (Copyright (c) 2011) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

Оглавление

1. Введение.....	2
1.1. Терминология.....	2
1.2. Принятая по умолчанию обработка.....	2
1.3. Управляемое клиентом извлечение принятых по умолчанию данных.....	3
2. Базовые режимы принятой по умолчанию обработки.....	3
2.1. Базовый режим report-all.....	3
2.1.1. Базовый режим извлечения report-all.....	3
2.1.2. Извлечение report-all <with-defaults>.....	3
2.1.3. Поведение report-all <edit-config> и <copy-config>.....	3
2.2. Базовый режим trim.....	3
2.2.1. Базовый режим извлечения trim.....	4
2.2.2. Извлечение trim <with-defaults>.....	4
2.2.3. Поведение trim <edit-config> и <copy-config>.....	4
2.3. Базовый режим explicit.....	4
2.3.1. Базовый режим извлечения explicit.....	4
2.3.2. Извлечение explicit <with-defaults>.....	4
2.3.3. Поведение explicit <edit-config> и <copy-config>.....	4
3. Извлечение принятых по умолчанию данных.....	4
3.1. Режим извлечения report-all.....	4
3.2. Режим извлечения trim.....	4
3.3. Режим извлечения explicit.....	4
3.4. Режим извлечения report-all-tagged.....	5
4. Свойство :with-defaults.....	5
4.1. Обзор.....	5
4.2. Зависимости.....	5
4.3. Идентификатор возможности.....	5

¹Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

²Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

4.4. Новые операции.....	5
4.5. Изменение имеющихся операций.....	5
4.5.1. Операции <get>, <get-config>, <copy-config>.....	5
4.5.2. Операция <edit-config>.....	6
4.5.3. Другие операции.....	6
4.6. Взаимодействие с другими возможностями.....	6
5. Модуль YANG для параметра <with-defaults>.....	6
6. XSD для атрибута default.....	8
7. Взаимодействие с IANA.....	8
8. Вопросы безопасности.....	9
9. Благодарности.....	9
10. Нормативные документы.....	9
Приложение А. Примеры использования.....	9
А.1. Пример модуля YANG.....	9
А.2. Пример набора данных.....	10
А.3. Примеры протокольных операций.....	10
А.3.1. <with-defaults> = report-all.....	10
А.3.2. <with-defaults> = report-all-tagged.....	11
А.3.3. <with-defaults> = trim.....	11
А.3.4. <with-defaults> = explicit.....	12

1. Введение

Протокол NETCONF [RFC6241] определяет способы чтения данных конфигурации и состояния с сервера NETCONF. Часть данных конфигурации клиент NETCONF может не устанавливать, используя вместо этого принятые по умолчанию значения. Во многих случаях клиент NETCONF заранее знает принятые по умолчанию данные и серверу NETCONF не требуется передавать их клиенту. Эти сведения могут быть получены, например, из документов, формально описывающих модели данных, поддерживаемые сервером NETCONF.

Клиенту важно точно знать, как реализация сервера обрабатывает принятые по умолчанию данные. В некоторых операциях протокола имеются тонкие различия, где принятое по умолчанию поведение сервера влияет на результат.

1.1. Терминология

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

Data model schema - схема модели данных

Документ или набор документов, описывающих поддерживаемые сервером NETCONF модели данных.

Management application - управляющее приложение

Компьютерная программа за пределами сервера NETCONF, которая настраивает или отслеживает сервер NETCONF. Управляющая программа может взаимодействовать с сервером, например, по протоколу NETCONF, через командный интерфейс (CLI) или простой протокол управления сетью (Simple Network Management Protocol или SNMP).

Schema default data - принятые по умолчанию данные схемы

Данные, заданные в схеме модели данных как принятые по умолчанию, т. е. устанавливаемые или используемые устройством всякий раз, когда клиент NETCONF или иное управляющее приложение (пользователь) не указывает конкретное значение для соответствующего узла данных. Принятые по умолчанию данные схемы могут сохраняться как часть хранилища конфигурации в зависимости от используемой сервером базовой модели.

Default data - принятые по умолчанию данные

Концептуальные данные, включающие принятое по умолчанию значение. Принятые по умолчанию данные не помещаются в хранилище. Не все серверы применяют одинаковые критерии для выбора узлов данных, помещаемых в хранилище. Если узла данных нет в хранилище и сервер вместо него использует принятое по умолчанию определение из схемы, узел считается заданным по умолчанию.

Default value - принятое по умолчанию значение

Значение экземпляра узла данных, концептуально применяемое сервером при отсутствии экземпляра этого узла.

Explicitly set data - явный набор данных

Данные, для которых клиент NETCONF или иное управляющее приложение устанавливает любое значение с помощью явной операции управления, включая принятое по умолчанию значение в схеме. Любое значение, установленное сервером NETCONF, которое не задано в схеме по умолчанию, также относится к явному набору данных.

<with-defaults> retrieval - извлечение <with-defaults>

Протокольная операция с параметром <with-default> для управления обработкой принятых по умолчанию данных.

:with-defaults

Сокращённое обозначение идентификатора возможности with-defaults.

Ниже перечислены термины, определённые в [RFC6241].

- client - клиент;
- datastore - хранилище данных;
- operation - операция;
- server - сервер.

Термин «узел данных» (data node) определён в [RFC6020].

1.2. Принятая по умолчанию обработка

Поведение default-handling, используемое сервером, влияет на протокольные операции NETCONF, как указано ниже.

1. Извлечение данных. Обычно серверу разрешается исключать узлы данных, которые считаются содержащими принятое по умолчанию значение. Фактически опускаемые узлы зависят от используемого сервером поведения принятой по умолчанию обработки.
2. Операции создания и удаления. Атрибут `<edit-config>` operation может служить для создания и/или удаления конкретных узлов данных. Эти операции зависят от наличия в данный момент целевого узла. Поведение сервера в части принятой по умолчанию обработки будет определять наличие запрашиваемого узла в конфигурационном хранилище.

1.3. Управляемое клиентом извлечение принятых по умолчанию данных

Сетевое устройство может иметь множество принятых по умолчанию значений. Зачастую такие значения конкретно определяются с разумными значениями, которые документированы и общеизвестны, так что пользователю не требуется их обрабатывать. По этой причине сетевые устройства достаточно часто не выводят параметры, имеющие принятые по умолчанию значения.

Однако в некоторых случаях клиенту NETCONF требуется получать от сервера принятые по умолчанию значения:

- управляющим приложениям часто нужен один определённый и полный набор конфигурационных значений, определяющих работу сетевого устройства;
- документация о принятых по умолчанию значениях может быть недоступна или ненадёжна;
- у некоторых управляющих приложений может не быть возможности корректно анализировать и интерпретировать формальные модели данных;
- пользователям может потребоваться понимание полученных данных без обращения к документации.

Во всех случаях клиенту NETCONF нужен механизм извлечения принятых по умолчанию данных с сервера NETCONF.

Документ определяет свойство протокола NETCONF для идентификации поведения сервера применительно к принятым по умолчанию данным, атрибут XML [W3C.REC-xmlschema-0-20041028] для идентификации принятых по умолчанию данных и расширение модуля YANG для протокола NETCONF, позволяющее клиенту NETCONF контролировать получение от сервера принятых по умолчанию данных.

2. Базовые режимы принятой по умолчанию обработки

Не все реализации серверов одинаково трактуют принятые по умолчанию данные. Вместо навязывания одной стратегии реализации этот документ разрешает серверам анонсировать конкретный стиль обработки принятых по умолчанию значений и клиент может подстроиться к такому поведению. Реализации клиентов предполагаются достаточно мощными для поддержки всех трёх режимов обработки принятых по умолчанию данных.

Серверы NETCONF возвращают принятые по умолчанию данные разными способами. Этот документ задаёт три стандартных базовых режима обработки принятых по умолчанию данных, которые может выбирать сервер:

- report-all (сообщать все);
- trim (подрезка);
- explicit (исключение).

Сервер **должен** выбрать один из трёх режимов, определённых здесь, для обработки принятых по умолчанию данных.

2.1. Базовый режим report-all

В режиме report-all сервер не считает данные принятыми по умолчанию, даже если они указаны такими в схеме.

2.1.1. Базовый режим извлечения report-all

При извлечении данных с сервера в режиме report-all и отсутствии параметра `<with-defaults>` **должны** возвращаться все узлы данных.

2.1.2. Извлечение report-all `<with-defaults>`

Если сервер использует базовый режим report-all, он должен поддерживать параметр `<with-defaults>` со значением report-all, как указано в параграфе 3.1.

2.1.3. Поведение report-all `<edit-config>` и `<copy-config>`

Сервер **должен** считать, что каждый узел данных существует, даже если он указан в схеме как принятый по умолчанию. Действительный атрибут операции create для узла данных, содержащий принятое по умолчанию в схеме значение, **должен** давать отказ с тегом ошибки data-exists. Действительный атрибут операции delete для узла данных, содержащего принятое по умолчанию в схеме значение, **должен** возвращать успех, даже если узел данные непосредственно заменяется сервером принятым по умолчанию значением.

Сервер, использующий базовый режим report-all, не имеет концепции принятого по умолчанию узла, поэтому режим извлечения report-all-tagged `<with-defaults>` не имеет смысла. Помеченных узлов не будет, поскольку нет узлов, которые пропускаются в операции базового режима извлечения. Если в каких-либо данных конфигурации присутствует атрибут default, сервер **должен** возвращать отклик `<rpc-error>` с тегом ошибки unknown-attribute.

2.2. Базовый режим trim

Сервер, использующий базовый режим trim, должен считать принятыми по умолчанию данными, которые указаны таковыми в схеме.

2.2.1. Базовый режим извлечения *trim*

При извлечении данных с сервера, использующего базовый режим *trim*, и отсутствии параметра `<with-defaults>` **недопустимо** сообщать узлы данных, если они содержат указанное в схеме значение по умолчанию. **Недопустимо** сообщать не являющиеся конфигурационными узлы данных, содержащие указанные в схеме данные по умолчанию.

2.2.2. Извлечение *trim* `<with-defaults>`

В базовом режиме *trim* сервер **должен** поддерживать параметр `<with-defaults>` со значением *trim*, как указано в параграфе 3.2.

2.2.3. Поведение *trim* `<edit-config>` и `<copy-config>`

Сервер **должен** считать существующим любой узел данных, который не содержит принятое в схеме по умолчанию значение. Действительный атрибут операции `create` для узла данных, указанного в схеме как принятый по умолчанию, **должен** приводить к успеху. Действительный атрибут операции `delete` для отсутствующего узла данных, указанного в схеме как принятый по умолчанию, **должен** вызывать отказ. Сервер **должен** возвращать отклик `<rpc-error>` с тегом ошибки `data-missing`.

Если клиент устанавливает для узла данных принятое в схеме по умолчанию значение с использованием любой действительной операции, это **должно** завершаться успехом, хотя узел данных **недопустимо** записывать в хранилище конфигурации NETCONF. Это имеет такой же эффект, как удаление узла данных и его трактовка как принятых по умолчанию данных.

Если сервер поддерживает значение `report-all-tagged` для параметра `<with-defaults>`, атрибут `default` **должен** приниматься на входе конфигурации, как указано в параграфах 4.5.1 и 4.5.2.

2.3. Базовый режим *explicit*

Сервер в базовом режиме *explicit* **должен** считать любой узел данных, не установленный явно, принятыми по умолчанию данными.

2.3.1. Базовый режим извлечения *explicit*

При извлечении данных с сервера в базовом режиме *explicit* и отсутствии параметра `<with-defaults>` узлы данных **должны** сообщаться, если они явно установлены клиентом, даже в случае заданного в схеме по умолчанию значения. Не относящиеся к конфигурации узлы данных с принятыми в схеме по умолчанию значениями **должны** включаться.

2.3.2. Извлечение *explicit* `<with-defaults>`

В базовом режиме *explicit* сервер **должен** поддерживать параметр `<with-defaults>` = *explicit* (параграф 3.3).

2.3.3. Поведение *explicit* `<edit-config>` и `<copy-config>`

Сервер считает существующими любые данные, установленные явно. Действительный атрибут операции `create` для узла данных, установленного клиентом в принятое схемой по умолчанию значение, **должен** приводить к отказу с тегом ошибки `data-exists`. Действительный атрибут операции `create` для узла данных, установленного сервером в принятое схемой по умолчанию значение, **должен** приводить к успеху. Действительный атрибут операции `delete` для узла данных, установленного клиентом в принятое схемой по умолчанию значение **должен** приводить к успеху. Действительный атрибут операции `delete` для узла данных, установленного сервером в принятое схемой по умолчанию значение, **должен** приводить к отказу с тегом ошибки `data-missing`.

Если сервер поддерживает режим извлечения `report-all-tagged` со свойством `:with-defaults`, атрибут `default` **должен** восприниматься при вводе конфигурации. Если параметры NETCONF `<edit-config>` или `<copy-config>` действительны, сервер будет считать помеченный узел данных (т. е. с атрибутом `default`, имеющим значение `true` или `1`) как запрос на возврат этого узла к принятому по умолчанию значению. Если этот запрос действителен в контексте операции NETCONF, узел данных удаляется и возвращается принятое по умолчанию значение. Узел данных в сообщении NETCONF **должен** содержать значение, которое в этом случае **должно** совпадать с принятым по умолчанию в схеме. В ином случае сервер **должен** возвращать отклик `<rpc-error>` с тегом ошибки `invalid-value`.

3. Извлечение принятых по умолчанию данных

Этот документ определяет новый параметр `<with-defaults>`, который может быть добавлен в запрос конкретной операции NETCONF для управления трактовкой сервером принятых по умолчанию данных.

Сервер, реализующий эту спецификацию, **должен** воспринимать параметр `<with-defaults>`, содержащий перечисляемое значение для поддерживаемого режима обработки заданных по умолчанию значений. Параметр `<with-defaults>` содержит одно из четырёх перечисляемых значений, определённых в этом разделе.

3.1. Режим извлечения *report-all*

При извлечении данных с `<with-defaults>` = *report-all* **должны** указываться все узлы данных, которые сервер считает принятыми по умолчанию.

3.2. Режим извлечения *trim*

При извлечении данных с `<with-defaults>` = *trim*, **недопустимо** указывать узлы данных, содержащие принятые в схеме по умолчанию значения. Узлы данных, не являющихся конфигурационными с принятыми в схеме по умолчанию значениями, указывать **недопустимо**.

3.3. Режим извлечения *explicit*

При извлечении данных с `<with-defaults>` = *explicit* узлы, для которых клиент установил принятое в схеме по умолчанию значение, **должны** включаться в отчёт. **Недопустимо** указывать концептуальные узлы данных, для которых сервер

установил принятое в схеме по умолчанию значение. Не относящиеся к конфигурации узлы с принятым в схеме по умолчанию значением **должны** указываться в отчёте.

3.4. Режим извлечения report-all-tagged

В дополнение к базовым режимам имеется специальный вариант режима report-all, названный report-all-tagged. Этот режим **должен** поддерживаться на сервере, если параметр also-supported в свойстве :with-defaults содержит опцию report-all-tagged. Детали кодирования для этого свойства приведены в разделе 4.

В этом режиме сервер возвращает все узлы данных подобно режиму report-all, за исключением того, что узлы, которые сервер считает содержащими принятые по умолчанию данные, будут включать атрибут XML для индикации этого условия. Это полезно для приложений, чтобы определить узлы, которые сервер считает содержащими принятые по умолчанию данные, в одной операции извлечения.

Сервер, поддерживающий report-all-tagged, **должен** также воспринимать атрибут XML default при вводе конфигурации в операции <edit-config> и <copy-config>. Детали представления XML для атрибута XML default приведены в разделе 6.

4. Свойство :with-defaults

4.1. Обзор

Свойство :with-defaults указывает базовый режим обработки принятых по умолчанию данных на сервере, а также может указывать поддержку дополнительных режимов извлечения принятых по умолчанию данных. Эти режимы извлечения позволяют клиентам NETCONF контролировать, какие из принятых по умолчанию данных возвращает сервер. Свойство влияет на данные конфигурации и состояния (использование принятых по умолчанию значений для состояния менее распространено). Отправка принятых по умолчанию данных контролируется отдельно для каждой операции.

Сервер NETCONF, реализующий свойство :with-defaults:

- **должен** указать свой базовый режим поведения параметром basic-mode в URI свойств (параграф 4.3);
- **должен** поддерживать модуль YANG из раздела 5 для режима обработки принятых по умолчанию данных, указанного параметром basic-mode;
- **следует** поддерживать модуль YANG из раздела 5 для режима обработки принятых по умолчанию данных, указанного значением report-all или report-all-tagged;
- при поддержке режима report-all-tagged **должен** поддерживать атрибут default;
- **может** поддерживать модуль YANG из раздела 5 для дополнительных режимов обработки принятых по умолчанию данных.

4.2. Зависимости

Нет.

4.3. Идентификатор возможности

```
urn:ietf:params:netconf:capability:with-defaults:1.0
```

Идентификатор **должен** иметь параметр basic-mode, указывающий, как сервер будет трактовать принятые по умолчанию данные, как указано в разделе 2. Параметр может принимать значения report-all, trim, explicit (раздел 2).

Идентификатор **может** иметь параметр also-supported, указывающий дополнительные перечисляемые значения (в дополнение к базовым режимам), которые сервер будет воспринимать для параметра <with-defaults> (раздел 5). Значением параметра является список разделённых запятыми режимов, которые поддерживаются в дополнение к режиму, указанному параметром basic-mode. Возможны значения report-all, report-all-tagged, trim, explicit (раздел 3).

Отметим, что URI этой возможности протокола отделен от URI свойства модуля YANG (раздел 5). Сервер, реализующий этот модуль, **должен** анонсировать URI возможности модуля YANG в соответствии с [RFC6020].

Ниже приведены примеры идентификаторов возможностей.

```
urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=explicit
urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=explicit&also-supported=report-all,report-all-tagged
```

4.4. Новые операции

Нет.

4.5. Изменение имеющихся операций

4.5.1. Операции <get>, <get-config>, <copy-config>

Новый элемент XML <with-defaults> добавляется на вход операций <get>, <get-config>, <copy-config>. При наличии элемента <with-defaults> он управляет возвратом принятых по умолчанию данных. Сервер должен возвращать принятые по умолчанию данные в сообщениях NETCONF <rpc-reply> в соответствии со значением этого элемента, если сервер поддерживает заданный режим извлечения. Параметр влияет лишь на указанные операции извлечения, не оказывая влияния на другие операции и энергонезависимые хранилища данных.

Элемент <with-defaults> определён в пространстве имён XML для модуля ietf-netconf-with-defaults.yang (раздел 5), а не в пространстве имён XML для операций <get>, <get-config>, <copy-config>. Разрешённые значения элемента with-defaults берутся из определения типа with-defaults-type в разделе 5. Разрешённые значения для конкретного сервера ограничены набором, указанным сервером как поддерживаемые в возможности with-defaults параметрами basic-mode и also-supported.

При использовании неподдерживаемого значения сервер NETCONF **должен** возвращать <rpc-error> с тегом ошибки invalid-value. Если элемент <with-defaults> отсутствует, сервер **должен** следовать базовому режиму, указанному параметром basic-mode в идентификаторе возможности :with-defaults, как указано в параграфе 4.3.

Операции <get> и <get-config> поддерживают отдельные механизмы фильтрации, используя параметр <filter>. Принятая по умолчанию фильтрация концептуально выполняется до обработки параметра <filter>. Например, при <with-defaults> = report-all параметр <filter> концептуально применяется ко всем узлам данных и всем данным, принятым по умолчанию. Параметр <with-defaults> влияет на операцию <copy-config> лишь при указании цели операции в параметре <url>. Если целью является хранилище конфигурации NETCONF (running, candidate, startup), параметр <with-defaults> не оказывает влияния. Сервер должен использовать свой базовый режим при копировании данных в хранилище конфигурации NETCONF. При наличии в этом случае параметра <with-defaults> сервер **должен** просто игнорировать его.

Если сервер поддерживает режим report-all-tagged, атрибут default, заданный в разделе 6, влияет также на операцию <copy-config>. При установке для атрибута default значения true или 1 сервер **должен** рассматривать новый узел данных как запрос на возврат узла к принятому по умолчанию значению (т. е. удаление из хранилища конфигурации). Узел данных в сообщении NETCONF **должен** в этом случае содержать значение, которое **должно** совпадать с принятым в схеме по умолчанию. В ином случае сервер должен возвращать отклик <rpc-error> с тегом invalid-value.

4.5.2. Операция <edit-config>

Операция <edit-config> имеет несколько режимов редактирования. На операции create и delete влияет базовый режим обработки принятых по умолчанию значений. На другие перечисляемые значения атрибута операции NETCONF влияния не оказывается.

Если атрибут операции содержит значение create и узел данных уже существует, сервер **должен** возвращать отклик <rpc-error> с тегом ошибки invalid-value.

Если клиент устанавливает для узла принятое по умолчанию в схеме значение, сервер **должен** воспринимать действительные запросы. Сервер **должен** сохранять или отбрасывать новое значение на основе базового режима обработки принятых по умолчанию значений. В режиме trim все принятые в схеме значения по умолчанию отбрасываются, в иных случаях представленные клиентом значения, принятые в схеме по умолчанию, записываются в хранилище конфигурации NETCONF.

Если сервер поддерживает режим report-all-tagged, атрибут default, заданный в разделе 6, влияет на операцию <edit-config>. При установке для атрибута default значения true или 1 сервер **должен** рассматривать новый узел данных как запрос на возврат узла к принятому по умолчанию значению (т. е. удаление из хранилища конфигурации). Узел данных в сообщении NETCONF **должен** в этом случае содержать значение, которое **должно** совпадать с принятым в схеме по умолчанию. В ином случае сервер должен возвращать отклик <rpc-error> с тегом invalid-value.

При наличии атрибута default эффективной операцией для целевого узла данных **должна** быть create, merge или герласе. В иных случаях сервер **должен** возвращать отклик <rpc-error> с тегом ошибки invalid-value. Например, при эффективной операции create запрос на создание должен быть действительным сам по себе (т. е., **недопустимо** наличие создаваемого узла). Процедура определения эффективной операции задана в [RFC6241]. Операция выводится из параметра default-operation и/или иных атрибутов операции, которые присутствуют в узле данных или любом из его предков в запросе <edit-config>.

4.5.3. Другие операции

Другим операциям, возвращающим данные конфигурации, также **следует** обрабатывать принятые по умолчанию данные в соответствии с правилами этого документа и явно указывать это в своей документации. Если это не указано в документе, определяющем операцию, описанные здесь правила обработки принятых по умолчанию значений не применяются к такой операции.

4.6. Взаимодействие с другими возможностями

Нет.

5. Модуль YANG для параметра <with-defaults>

Ниже приведён модуль YANG, определяющий добавление элемента with-defaults к операциям <get>, <get-config> и <copy-config>. Язык YANG определён в [RFC6020], указанные операции определены для YANG в [RFC6241]. Каждый сервер NETCONF, поддерживающий :with-defaults, должен реализовать этот модуль YANG.

```
<CODE BEGINS> file="ietf-netconf-with-defaults@2011-06-01.yang"
module ietf-netconf-with-defaults {

    namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults";

    prefix ncwd;

    import ietf-netconf { prefix nc; }

    organization
        "IETF NETCONF (Network Configuration Protocol) Working Group";

    contact
        "WG Web: <http://tools.ietf.org/wg/netconf/>

        WG List: <netconf@ietf.org>

        WG Chair: Bert Wijnen <bertietf@bwiijnen.net>

        WG Chair: Mehmet Ersue <mehmet.ersue@nsn.com>
```

Editor: Andy Bierman <andy.bierman@brocade.com>

Editor: Balazs Lengyel <balazs.lengyel@ericsson.com>;

description

"Модуль задаёт расширение протокола NETCONF, позволяющее клиентам NETCONF управлять обработкой принятых по умолчанию значений на сервере в конкретных операциях NETCONF.

Авторские права (Copyright (c) 2011) принадлежат IETF Trust и лицам, указанным как авторы. Все права защищены.

Распространение и использование в исходной или двоичной форме в изменениями или без таковых разрешено в соответствии с условиями, указанными в Simplified BSD License раздела 4.с IETF Trust's Legal Provisions применительно к документам IETF (<https://trustee.ietf.org/license-info>).

Эта версия модуля является частью RFC 6243, где правовые аспекты изложены более полно.";

revision 2011-06-01 {

description

"Исходная версия.";

reference

"RFC 6243: With-defaults Capability for NETCONF";

}

typedef with-defaults-mode {

description

"Возможные режимы возврата принятых по умолчанию данных.";

reference

"RFC 6243; Section 3.";

type enumeration {

enum report-all {

description

"Указываются все принятые по умолчанию данные.";

reference

"RFC 6243; Section 3.1";

}

enum report-all-tagged {

description

"Указываются все принятые по умолчанию данные. Все узлы, сочтённые принятыми по умолчанию даные, будут содержать атрибут XML default со значением true или 1.";

reference

"RFC 6243; Section 3.4";

}

enum trim {

description

"Значения не возвращаются, если они приняты по умолчанию.";

reference

"RFC 6243; Section 3.2";

}

enum explicit {

description

"Возвращаются значения, содержащие определение явно установленных данных.";

reference

"RFC 6243; Section 3.3";

}

}

grouping with-defaults-parameters {

description

"Параметр <with-defaults> для управления операциями NETCONF при извлечении принятых по умолчанию данных.";

leaf with-defaults {

description

"Запрос режима обработки явных значений по умолчанию.";

reference

"RFC 6243; Section 4.5.1";

type with-defaults-mode;

}

}

// Расширение операции get-config

augment /nc:get-config/nc:input {

description

"Добавляет параметр <with-defaults> на вход операции NETCONF <get-config>.";

reference

"RFC 6243; Section 4.5.1";

uses with-defaults-parameters;

```

}

// Расширение операции get
augment /nc:get/nc:input {
  description
    "Добавляет параметр <with-defaults> на вход операции
    NETCONF <get>.";
  reference
    "RFC 6243; Section 4.5.1";

  uses with-defaults-parameters;
}

// Расширение операции copy-config
augment /nc:copy-config/nc:input {
  description
    "Добавляет параметр <with-defaults> на вход операции
    NETCONF <copy-config>.";
  reference
    "RFC 6243; Section 4.5.1";

  uses with-defaults-parameters;
}
}
<CODE ENDS>

```

6. XSD для атрибута default

Приведённый ниже документ XML Schema [W3C.REC-xml-20081126] определяет атрибут default, описанный в этом документе. XSD применяется лишь при поддержке сервером режима report-all-tagged обработки принятых по умолчанию значений.

Атрибут default использует тип данных XSD boolean. В соответствии с параграфом 3.2.2.1 XML Schema Part 2: Datatypes, допустимым лексическим представлением типа xs:boolean являются строки 0 и false для значения false и 1 или true - для true. реализации **должны** поддерживать оба варианта представления.

```

<CODE BEGINS> file="defaults.xsd"
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:netconf:default:1.0"
  targetNamespace="urn:ietf:params:xml:ns:netconf:default:1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xml:lang="en">

  <xs:annotation>
    <xs:documentation>
      Эта схема определяет синтаксис атрибута default, заданного в
      этом документе.
    </xs:documentation>
  </xs:annotation>

  <!-- Атрибут default -->
  <xs:attribute name="default" type="xs:boolean" default="false">
    <xs:annotation>
      <xs:documentation>
        Этот атрибут указывает, рассматривается ли сервером представленный
        этим элементом XML атрибут как принятые по умолчанию данные. При
        установке true или 1 узел данных считается принятым по умолчанию.
        Значение false или 0 говорит, что узел данных не принят по умолчанию.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:schema>
<CODE ENDS>

```

7. Взаимодействие с IANA

Этот документ регистрирует URN идентификатора возможности в реестре Network Configuration Protocol (NETCONF) Capability URNs

```
urn:ietf:params:netconf:capability:with-defaults:1.0
```

Документ регистрирует два URN пространства имён XML в реестре IETF XML в соответствии с форматом [RFC3688].

```
URI: urn:ietf:params:xml:ns:netconf:default:1.0
```

```
URI: urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults
```

```
Registrant Contact: The NETCONF WG of the IETF.
```

```
XML: N/A, the requested URIs are XML namespaces.
```

Документ регистрирует имя модуля в реестре YANG Module Names, определённом в [RFC6020].

```
name: ietf-netconf-with-defaults
```

```
prefix: ncwd
```

```
namespace: urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults
```

```
RFC: 6243
```

8. Вопросы безопасности

Документ задаёт расширение имеющихся операций протокола NETCONF. Документ не создаёт новых и не повышает имеющиеся риски безопасности для систем управления.

Свойство with-defaults даёт клиентам средство управления извлечением принятых по умолчанию данных из хранилищ NETCONF. Вопросы безопасности, отмеченные в [RFC6241], применимы и здесь.

9. Благодарности

Спасибо Martin Bjorklund, Sharon Chisholm, Phil Shafer, Juergen Schoenwaelder, Kent Watsen, Washam Fan и другим участникам NETCONF WG за важный вклад в этот документ.

10. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), January 2004.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011.
- [W3C.REC-xml-20081126] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<http://www.w3.org/TR/2008/REC-xml-20081126>>.
- [W3C.REC-xmlschema-0-20041028] Fallside, D. and P. Walmsley, "XML Schema Part 0: Primer Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-0-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-0-20041028>>.

Приложение А. Примеры использования

А.1. Пример модуля YANG

Ниже приведён пример модуля YANG с таблицей интерфейсов для демонстрации поведения параметра <with-defaults> с конкретной моделью данных.

Отметим, что это просто пример и реализация модуля не требуется для поддержки свойства with-defaults, заданного в раздел 4. Модуль не регистрируется в IANA и не считается компонентом кода. Модуль преднамеренно краток и включает лишь несколько описательных операторов.

```

module example {
  namespace "http://example.com/ns/interfaces";

  prefix exam;

  typedef status-type {
    description "Статус интерфейса";
    type enumeration {
      enum ok;
      enum 'waking up';
      enum 'not feeling so good';
      enum 'better check it out';
      enum 'better call for help';
    }
    default ok;
  }

  container interfaces {
    description "Пример группы интерфейсов";

    list interface {
      description "Пример записи для интерфейса";
      key name;

      leaf name {
        description
          "Административное имя интерфейса. Этот
          идентификатор уникален лишь в зоне действия
          списка на конкретном сервере.";
        type string {
          length "1 .. max";
        }
      }

      leaf mtu {
        description
          "Значение MTU для интерфейса.";
        type uint32;
        default 1500;
      }
    }
  }
}

```

```

    leaf status {
      description
        "Текущий статус интерфейса.";
      type status-type;
      config false;
    }
  }
}

```

A.2. Пример набора данных

Приведённый ниже элемент данных показывает концептуальное содержимое примера сервера для протокольных операций из следующего параграфа. Пример включает все узлы данных конфигурации, не являющиеся конфигурационными узлы и принятые по умолчанию листья.

```

<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="http://example.com/ns/interfaces">
    <interface>
      <name>eth0</name>
      <mtu>8192</mtu>
      <status>up</status>
    </interface>
    <interface>
      <name>eth1</name>
      <mtu>1500</mtu>
      <status>up</status>
    </interface>
    <interface>
      <name>eth2</name>
      <mtu>9000</mtu>
      <status>not feeling so good</status>
    </interface>
    <interface>
      <name>eth3</name>
      <mtu>1500</mtu>
      <status>waking up</status>
    </interface>
  </interfaces>
</data>

```

В этом примере поле `mtu` для каждой интерфейсной записи указано в таблице.

	<i>name</i>	<i>setby</i>	<i>mtu</i>
eth0		client	8192
eth1		server	1500
eth2		client	9000
eth3		client	1500

A.3. Примеры протокольных операций

Приведённые ниже примеры показывают некоторые операции `<get>` с использованием элемента `with-defaults`. Используемая модель данных определена в Приложении A.1.

Клиент извлекает все узлы данных объекта `interfaces`, фильтруя их по параметру `<with-defaults>`.

A.3.1. `<with-defaults> = report-all`

В этом примере иллюстрируется обработка параметра `<with-defaults>` в режиме `report-all`.

```

<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <interfaces xmlns="http://example.com/ns/interfaces"/>
    </filter>
    <with-defaults
      xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults">
      report-all
    </with-defaults>
  </get>
</rpc>

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="http://example.com/ns/interfaces">
      <interface>
        <name>eth0</name>
        <mtu>8192</mtu>
        <status>up</status>
      </interface>
      <interface>
        <name>eth1</name>
        <mtu>1500</mtu>
        <status>up</status>
      </interface>
      <interface>

```

```

    <name>eth2</name>
    <mtu>9000</mtu>
    <status>not feeling so good</status>
  </interface>
  <interface>
    <name>eth3</name>
    <mtu>1500</mtu>
    <status>waking up</status>
  </interface>
</interfaces>
</data>
</rpc-reply>

```

A.3.2. <with-defaults> = report-all-tagged

В этом примере иллюстрируется обработка параметра <with-defaults> в режиме report-all-tagged. Помеченными (tagged) узлами данных являются элементы, содержащие атрибут XML default со значением true или 1.

Фактически помечаемые сервером узлы данных зависят от базового режима обработки принятых по умолчанию значений на сервере. Помечаются лишь данные, которые сочтены принятыми по умолчанию.

В примере сервер использует базовый режим trim, поэтому помечаются все узлы данных с установленными в схеме по умолчанию значениями. В базовом режиме explicit, помечаются лишь узлы данных, которые не установлены явно, а в режиме report-all никакие узлы данных не помечаются.

```

<rpc message-id="102"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <interfaces xmlns="http://example.com/ns/interfaces"/>
    </filter>
    <with-defaults
      xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults">
      report-all-tagged
    </with-defaults>
  </get>
</rpc>

<rpc-reply message-id="102"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:wd="urn:ietf:params:xml:ns:netconf:default:1.0">
  <data>
    <interfaces xmlns="http://example.com/ns/interfaces">
      <interface>
        <name>eth0</name>
        <mtu>8192</mtu>
        <status wd:default="true">up</status>
      </interface>
      <interface>
        <name>eth1</name>
        <mtu wd:default="true">1500</mtu>
        <status wd:default="true">up</status>
      </interface>
      <interface>
        <name>eth2</name>
        <mtu>9000</mtu>
        <status>not feeling so good</status>
      </interface>
      <interface>
        <name>eth3</name>
        <mtu wd:default="true">1500</mtu>
        <status>waking up</status>
      </interface>
    </interfaces>
  </data>
</rpc-reply>

```

A.3.3. <with-defaults> = trim

В этом примере иллюстрируется обработка параметра <with-defaults> в режиме trim.

```

<rpc message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <interfaces xmlns="http://example.com/ns/interfaces"/>
    </filter>
    <with-defaults
      xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults">
      trim
    </with-defaults>
  </get>
</rpc>

<rpc-reply message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="http://example.com/ns/interfaces">

```

```

<interface>
  <name>eth0</name>
  <mtu>8192</mtu>
</interface>
<interface>
  <name>eth1</name>
</interface>
<interface>
  <name>eth2</name>
  <mtu>9000</mtu>
  <status>not feeling so good</status>
</interface>
<interface>
  <name>eth3</name>
  <status>waking up</status>
</interface>
</interfaces>
</data>
</rpc-reply>

```

A.3.4. **<with-defaults> = explicit**

В этом примере иллюстрируется обработка параметра <with-defaults> в режиме explicit.

```

<rpc message-id="104"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <interfaces xmlns="http://example.com/ns/interfaces"/>
    </filter>
    <with-defaults
      xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults">
      explicit
    </with-defaults>
  </get>
</rpc>

<rpc-reply message-id="104"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="http://example.com/ns/interfaces">
      <interface>
        <name>eth0</name>
        <mtu>8192</mtu>
        <status>up</status>
      </interface>
      <interface>
        <name>eth1</name>
        <status>up</status>
      </interface>
      <interface>
        <name>eth2</name>
        <mtu>9000</mtu>
        <status>not feeling so good</status>
      </interface>
      <interface>
        <name>eth3</name>
        <mtu>1500</mtu>
        <status>waking up</status>
      </interface>
    </interfaces>
  </data>
</rpc-reply>

```

Адреса авторов

Andy Bierman
 Brocade
 EMail: andy.bierman@brocade.com

Balazs Lengyel
 Ericsson
 Budapest,
 Hungary
 EMail: balazs.lengyel@ericsson.com

Перевод на русский язык

Николай Малых
nmalykh@protokols.ru